ANALYSIS OF S-CIRCUIT UNCERTAINTY

An Honors Fellows Thesis

by

TAAHIR DAVID AHMED

Submitted to the Honors Programs Office
Texas A&M University
in partial fulfillment of the requirements for the designation as

HONORS UNDERGRADUATE RESEARCH FELLOW

APRIL 2011

Majors: Computer Engineering
Physics

ANALYSIS OF S-CIRCUIT UNCERTAINTY

An Honors Fellows Thesis

by

TAAHIR DAVID AHMED

Submitted to the Honors Programs Office
Texas A&M University
in partial fulfillment of the requirements for the designation as

HONORS UNDERGRADUATE RESEARCH FELLOW

Approved by:

Research Advisor: Dylan Shell
Associate Director of Honors and Undergraduate Research: Dave A. Louis

APRIL 2011

Majors: Computer Engineering
Physics

ABSTRACT

Analysis of S-Circuit Uncertainty. (April 2011)

Taahir David Ahmed
Department of Electrical and Computer Engineering
Department of Physics
Texas A&M University

Research Advisor: Dr. Dylan Shell
Department of Computer Science and Engineering

The theory of sensori-computational circuits provides a capable framework for the description and optimization of robotic systems, including on-line optimizations. This theory, however, is inadequate in that it does not account for uncertainty in a robot's environment, sensing, and control. In order to allow meaningful optimization of robotic systems, a method for estimating the output uncertainty of an s-circuit in terms of the uncertainty in its input is formulated. This method is expressly designed to have low information requirements, to ensure that it is feasible to apply in practice. The method is subjected to experimental verification on two representational s-circuits, which confirms the validity of its predictions.

DEDICATION

To God, gracious and merciful, who has given me a good life and good faculties.

To Kate, who put up with me and offered words of encouragement long after most would have given up in frustration.

ACKNOWLEDGMENTS

I owe great thanks to Dr. Dylan Shell for giving me a place to work and a problem to ponder. Without his keen insight and steadfast encouragement, I would have lost my way time and time again. In addition, I must thank the members of the Multi-Robot Systems Lab, particularly Benjamin Fine and Shawn Kristek, for listening to my presentations, offering feedback, and putting up with my endless rambling about s-circuits. Thanks goes to Dr. Sarmad Adnan, who offered practical advice on the implementation of the beacon hardware. As well, I would like to thank Dr. Louis and the Department of Honors and Undergraduate Research as a whole for their support throughout my thesis process. In addition, I must thank to my colleagues, the Sophomore Advisors in McFadden and Lechner Halls, who have tolerated without complaint my lack of time to devote to our projects, and my fellow student Nick Mai for his commiseration. Finally, I would like to thank the faculty and staff of the Department of Computer Science and Engineering, the Department of Electrical and Computer Engineering, and the Department of Physics for giving me the mental and physical resources I need to investigate my world.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION: ON UNCERTAINTY AND S-CIRCUITS

Robots and robotic systems are ubiquitous in the modern world. Beyond the striking images of industrial manipulators and humanoid autonoma, any system that gathers data about the physical world and then acts upon it can be analyzed from a robotic perspective. Systems as widely varied as automatic doors and anti-lock brakes fall under this definition.

A robot uses both sensors and built-in knowledge to construct and act upon a model of its world. This model could be static and extremely simplistic, as is the case for many electromechanical toys; a robotic system need not even include a computer. At the other extreme, a robot might form a complex, dynamic model of its environment with an array of sensors and algorithms, a strategy frequently seen in autonomous vehicles. While these cases might not appear to have much in common, a robot's perception of the world is always complicated by the presence of uncertainty.

I-1 Uncertainty

In a perfect situation, there is a precise and accurate relationship between a robot's perception and its true environment. In practice, a robot encounters uncertainty in its sensor inputs, its model of the world, and its actuator outputs. This uncertainty causes the robot's perception of the environment to diverge from the truth. This can cause the robot to per-

---

This thesis follows the style and format of Artificial Intelligence.

form actions that, while correct under the robot's model of the environment, are incorrect in the true environment. Accounting for uncertainty is one of the fundamental challenges of robotics; as Thrun *et al.* relate [10], there have been three successive approaches to the issue.

Early work in robotics typically followed a model-based paradigm, in which the robot maintains an internal model of its environment that it trusts completely. This approach is not robust to uncertainty – experimenters often dealt with uncertainty by attempting to minimize it to the point where it could be reasonably ignored. Typically, this was accomplished by machining the robot to extremely tight tolerances and carefully controlling its environment — in effect, trying to spend away the problem. This approach can be quite expensive, and cannot be easily applied to robots that must operate in uncontrolled environments, such as public spaces or unexplored terrain. Historically, this approach was necessary because of severe limitations on computing power and availability in the early days of robotics.

As a backlash against the fragility of the model-based paradigm in the face of noise, focus shifted to a behavior-based paradigm. Robots maintained at most a minimal model of their environment, with complex behavior being produced by reaction to external stimuli. Behavior-based robots rely primarily on immediately sensable information, their applications are often limited to simple tasks where the requisite information is immediate and obvious.

As robots began to enter task domains that were more complex and required greater perceptive capabilities on the part of the robot, hybrid, uncertainty-based models began to be developed and adopted. Modern robots using a probabilistic paradigm often use both complex internal models and diverse sensing capabilities, but trust neither completely. Instead,

probabilistic robots are explicitly assume that their sensory inputs, internal models, and actuator outputs are uncertain, using statistical techniques to determine the correct action to take. This approach leads to robust performance in the face of environmental, sensory, and model-related uncertainty. Because this method carries comparatively high computational costs, the availability of greater computational power for cheaper prices over the last two decades has greatly contributed to its widespread adoption.

Sources of uncertainty

Cox and Leonard note that the uncertainty in a robotic system may be partitioned into value uncertainty and origin uncertainty [1]. Value uncertainty is the well-known phenomenon of jitter or noise in sensors or actuators — a sensor does not tend to report a single value for the quantity it measures, but rather a random distribution of values. Origin uncertainty describes the fact that a measurement that the robot makes could arise from any number of discrete objects in the robot's environment.

In addition, many common operations on data introduce artifacts (unwanted, nonrandom structures) into the data. Quantization, or binning, is a ubiquitous artifact that arises when a continuous quantity is sampled into a discrete representation. This process discards a great deal of information — it is impossible to reconstruct the original continuous quantity from its discrete approximation. Donald and Jennings investigate the effects of quantization on robotic systems, noting that it partitions a robot's state space into a finite number of perceptual equivalence classes [3]. States falling into the same perceptual equivalence class are indistinguishable to the robot, which Donald and Jennings note has the effect of causing some problem-solving strategies which are provably correct in a continuous state space to fail in a quantized state space. Another common form of artifact arises

when data is compressed, especially when frequency-based compression is used. Discrete cosine transform compression methods such as JPEG (for still images) and MJPEG (for video) discard higher-frequency components, resulting in very noticeable artifacts in the data. Artifacts must be considered with care, as their nonrandom nature often violates the assumptions of independence required by many statistical procedures. However, they are not explicitly considered in this work.

Modeling uncertainty

A robot's perception of uncertainty is often based upon Bayes' theorem, equation (1), which provides a method for incorporating new evidence into a previous estimate.

$$
\begin{aligned}
H &\equiv \text{hypothesis of interest} \\
E &\equiv \text{some new evidence} \\
\mathrm{P}(H) &\equiv \text{prior probability of } H \text{ before } E \text{ was observed} \\
\mathrm{P}(E) &\equiv \text{marginal probability of observing } E \text{ under all hypotheses} \\
\mathrm{P}(E|H) &\equiv \text{conditional probability of observing } E \text{ under } H \\
\mathrm{P}(H|E) &\equiv \text{posterior probability of } H \text{ after } E \text{ was observed} \\
\mathrm{P}(H|E) &= \frac{\mathrm{P}(E|H)\,\mathrm{P}(H)}{\mathrm{P}(E)}
\end{aligned}
\tag{1}
$$

The hypothesis and evidence are represented by belief distributions which are probability distributions mapping a particular state that the robot can take on to the probability that it is the true state.

Normal distributions are a popular representation of robotic belief distributions due in large part to their computational tractability — normal distributions are linear (the sum of

normal distributions follows a normal distribution) and compact (normal distributions are fully described by their mean and variance). Additionally, the uncertainty in the output of many sensors is approximately normal. Normal distributions do have a significant limitation in that they are unimodal, precluding belief distributions centered around more than one state. In practice, normal distributions are a sufficient description of belief in many situations. The Kalman filter is a well-established and computationally efficient Bayesian predictor that uses normal distributions to represent belief [5].

If a multimodal belief is required, a grid-based approach is typically used to approximate the continuous belief distribution. The state space is partitioned into regular intervals, with each interval being assigned an associated probability. The beliefs from the sensors are then combined, partition by partition, with the robot's prior belief. Grid-based approaches are, in general, far more computationally intensive than normal representations, though not prohibitively so. A detailed investigation of the performance of various grid-based Bayesian filters by Fox *et al.* can be found in [4].

Uncertainty in compound spatial relationships

Nearly ubiquitously in robotic systems, data is transformed several times before it is used. Because the data contains some describable uncertainty, it is natural to wonder what the relationship between the uncertainty in the transformed data and the uncertainty in the source data is. Smith and Cheeseman develop a solution to this problem in the context of robots moving through compound reference frames in a plane environment [8]. Smith, Self, and Cheeseman give a more general form of this result, equation (2), in the context

of a robot building a map of the features present in its environment [9].

$$\mathrm{Var}\left(f(\mathbf{x})\right) = \left[\frac{\partial f}{\partial \mathbf{x}}\right] \mathrm{Var}\left(\mathbf{x}\right) \left[\frac{\partial f}{\partial \mathbf{x}}\right]^{\mathrm{T}} \Bigg| \; f \in \theta(x) \tag{2}$$

An explanation of these notation conventions can be found in section II-2. It should be noted that this result makes no assumptions about the distribution of $\mathbf{x}$ — it holds for any distribution as long as the f is a linear function.

Equation (2) is significant because it is applicable for general linear functions f, not simply for those relating to spatial transformations. Functions that are nonlinear must be linearized about the mean of their inputs, an approximation justifiable in many but not all situations.

I-2 S-Circuits

Because robots are such a broad class of devices, it is often unclear what comparing any two robots means. It makes little sense to compare an unmanned rover to an industrial welding robot. However, when considering a set of robots that perform the same task, the ability to rank them based on their predicted performance is invaluable. O'Kane and LaValle present a mathematical framework that accomplishes this using a model of a robot similar to a Turing machine (a theoretical model of a general computer) [6]. They consider a robot to be defined by a current state X, a set of possible actions U, a set of possible observations Y, a state transition function $f : X \times U \to X$, and an observation function $h : X \times U \to Y$. From this, the most desirable state a robot can reach may be determined, which is sufficient information to rank a set of robots. While useful, this method suffers from significant shortcomings:

- Not all robots and tasks can easily be described with this model. In particular, it is well suited for describing robots following a behavior-based paradigm. It is less clear how to describe a robot that actively models its environment.

- This method ranks robots by *power*, not by *optimality*. If robot A can outperform robot B, but only at significant expense, then it is not clear that A can be considered better than B.

- This method incorporates only a limited treatment of uncertainty.

Donald presents an alternative approach to the comparison of robotic systems in his development of a theory of sensori-computational circuits [2]. This theory models the flow of information through a computational system composed of *modules* that are distributed across some physical environment. Modules represent some abstract sensory or computational capability with a specific location in the physical environment, for example a laser rangefinder or an FFT algorithm running on a microprocessor. Inter-module communication occurs via *datapaths*, which might represent physical communication channels such as network cables or computational channels such as a C return value. In effect, an s-circuit is a directed graph of module and datapath information flow embedded into its environment.

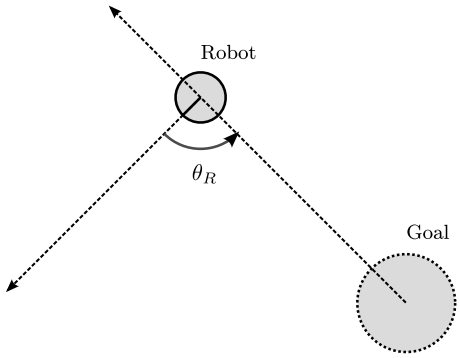
This embedding property gives rise to some unexpected theoretical results. Donald demonstrates two ways to extend the functionality of an s-circuit. He shows that s-circuits are modular — they may be *combined* to make a larger s-circuit. In addition, he notes that *permutation*, or a change of embedding, can radically change the behavior and power of an s-circuit.

Strengths

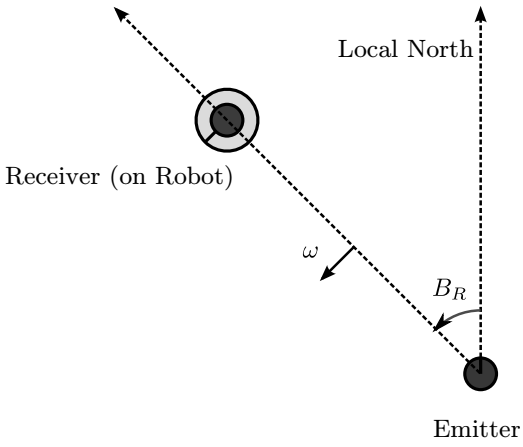
Donald applies his formulation of s-circuits to the task of finding the optimal sensor configuration for a given task. He introduces the notion of an *minimal task sensor*, the (perhaps fictional) sensor that provides the minimum information necessary to accomplish a given task. Given a catalog of s-circuit primitives to combine and permute (which might include singular modules and datapaths or more complex s-circuits), it is possible to determine the cheapest implementation of the minimal task sensor, where cost is some user-defined function. This is a powerful result, as it allows a formal, rigorous optimization against relevant criteria at the design phase of a robotic system.

Donald develops a motivational example that will be further explored in this work. He presents the goal-finding minimal task sensor (Figure 1(a)), which is the minimal task sensor for a plane robot attempting to move to a specified goal region. The sole information that the robot requires is the relative heading to the goal θ_R. By implementing a control loop in which the robot repeatedly turns by θ_R and drives forward some distance, the robot will eventually enter the goal region. To implement this minimal task sensor, Donald provides a catalog of three primitive s-circuit elements: a beacon, a compass, and communication.

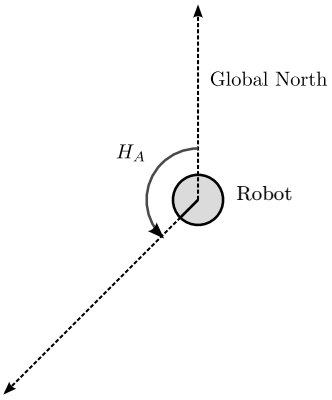
The beacon, detailed in Figure 1(b), is a sensor system composed of an emitter and a receiver. The emitter mounts two distinguishable lights — a rotating (with constant angular velocity) directional light that projects a ray in the plane, and an omnidirectional light that flashes when the rotating light crosses local North. The receiver measures the elapsed time between the omnidirectional flash and the flash when the rotating light points at the receiver. This information is sufficient to determine the relative bearing from the emitter to the receiver, B_R.



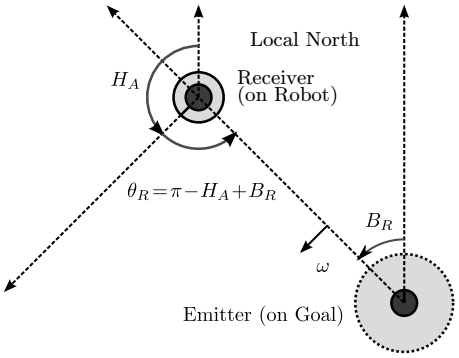
(a) Goal-finding minimal task sensor as given by Donald.



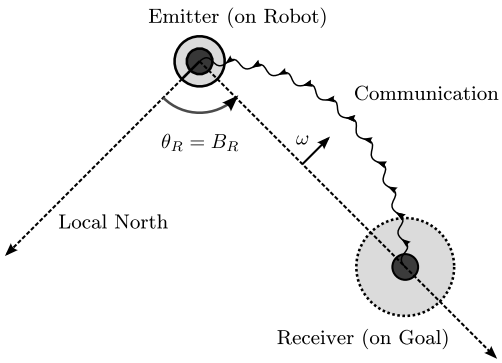
(b) Beacon sensor that returns bearing from emitter to receiver.



(c) Compass the returns absolute heading.



(d) Implementation of (a) with beacon-compass combination.



(e) Implementation of (a) with beacon permutation

Figure 1: Explanatory figures for Donald's s-circuit example.

The compass, detailed in Figure 1(c), reports the absolute heading with respect to global North. In the case of a magnetic compass, global North is fixed at the actual magnetic North. For gyroscopic or accelerometric compasses, global North may be any arbitrary reference direction.

The communication primitive is a datapath that allows one module to communicate its output to another. Its nature is unspecified, but in the context of this problem, it most likely represents a wireless or radio link.

From these primitives, Donald presents two possible constructions that simulate the minimal task sensor by providing θ_R. the first construction, hereafter referred to as the beacon-compass reconfiguration, is illustrated in Figure 1(d). The beacon emitter is placed at the center of the goal region (with its local North aligned to the global North), and both the receiver and the compass are bolted to the robot. Under this embedding of the modules,

$$\theta_R = \pi - H_A + B_R.$$

The other reconfiguration that Donald presents, hereafter referred to as the beacon-only reconfiguration, uses the beacon module and communication datapath. The beacon is embedded in the opposite configuration — the emitter is placed on the robot (local North aligned with the robot) and the receiver is placed at the center of the goal region. In this context,

$$\theta_R = B_R.$$

The only problem with this solution is that θ_R is known only at the receiver, not at the robot. The communication primitive solves this problem by allowing the information to

be transmitted to the robot.

Once the two reconfigurations have been developed, a user can determine which is cheaper given their specific cost for each primitive. However, because the treatment of s-circuits as Donald develops them does not account for uncertainty, it cannot be relied upon to provide the true best reconfiguration for a given situation. This is because it is possible that the cheaper s-circuit might display an unacceptable level of uncertainty in its output. In the remainder of this work, we propose and verify a technique that will allow a user to optimize s-circuit output uncertainty against cost.

I-3 Description of work

In this work, we extend Donald's s-circuit calculus to include a limited prediction of the uncertainty within an s-circuit. Smith, Self, and Cheeseman's formulae are used to form a model of the s-circuit using linear and approximately linear relationships. We then empirically test this model on example s-circuits given by Donald in his work.

A theory of s-circuits that provides a quantitative, empirically valid mode of reasoning about sensors and their outputs has enormous potential for improvements in the optimization of sensori-computational systems, both in robotics and in other disciplines. In addition to the optimization of robotic and sensor systems in their design phase, there is a possibility for run-time optimizations. Some possible applications are:

- An autonomous exploratory rover needs to navigate its environment with minimal power usage. The calculus provides an optimal rule set for it to determine whether or not it needs to activate auxiliary sensors to plot a more efficient route.

- A set of cooperative warehouse robots needs to be as cheap, simple, and reliable

as possible. The calculus is used to determine the absolute minimum number of sensors needed on each robot and in the environment.

- An autonomous car can be designed with the optimal set of sensors to take advantage of features that already exist on roadways, thus allowing a large step forward in technology while leaving the infrastructure it uses unchanged.

Automated systems that require input from sensors to make decisions and carry out tasks are becoming ubiquitous in society. A quantiable method for optimizing sensor systems will lead to benets for society at large by increasing the efficiency of both new and existing automated systems without requiring new technology, only better utilization of existing technology.

CHAPTER II

UNCERTAINTY IN S-CIRCUITS

II-1 Why track uncertainty?

As presented by Donald, the concept of s-circuits currently do not account for uncertainty [2]. An important potential application of s-circuit modeling is finding the least-cost implementation of a minimal task sensor. In any practical s-circuit, noise will be present and will typically be inversely correlated with the cost of the system. If a user needs to choose a meaningful best choice from several possible s-circuit implementations of a minimal task sensor, he or she must have an estimate of the output uncertainty of each s-circuit. With such an estimate, the problem becomes one of balancing output uncertainty against cost; without one, it is mere guesswork.

At the same time, there are limitations on the amount of information available to make a prediction of the output uncertainty. If a user has a catalog of modules and datapaths to work with, it is reasonable to assume that he or she can take basic measurements of the properties of each individual component, including calibration error and a few statistical moments. It is not reasonable to require the user to take measurements between every pair of components — the number of required measurements would be factorial in the size of the catalog. We will develop methods for making credible estimates of s-circuit output uncertainty using a single-component measurement, the variance.

II-2 Terminology, conventions, and notation

This work deals with many mathematical constructs that are often expressed using overlapping notation. To ensure that the formulae we present in this work are well-defined, we will clarify the notation for each concept that will be used in this work.

Matrices are denoted by bold uppercase ($\mathbf{A}$) and column vectors by bold lowercase ($\mathbf{a}$). Scalar quantities are denoted by non-bold symbols (a). When working with covariance matrices and multivariate functions, it is expedient to concatenate matrices, an operation denoted by a matrix of matrices.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_{11}} & \cdots & \mathbf{A_{1n}} \\ \vdots & \ddots & \vdots \\ \mathbf{A_{m1}} & \cdots & \mathbf{A_{mn}} \end{bmatrix}$$

This construct is often known as a block matrix.

If $\mathbf{x}$ is a random vector, then $\bar{\mathbf{x}}$ represents its mean, and $\tilde{\mathbf{x}}$ represents its residual, $(\mathbf{x} - \bar{\mathbf{x}})$.

When representing Jacobian matrices, the canonical form will be used. However, for compactness, all conditions on Jacobian matrices will be omitted:

$$\left[\frac{\partial f}{\partial \mathbf{x}}\right] \equiv \left[\frac{\partial f}{\partial \mathbf{x}}\right]_{\mathbf{x}=\bar{\mathbf{x}}}.$$

Thus, all Jacobians discussed in this work are constant quantities.

Second-order moments are used frequently in this work. The most general of these is

vector covariance, given by

$$\begin{aligned}\mathrm{Cov}\left(\mathbf{x},\mathbf{y}\right) &= \mathrm{E}\left[\tilde{\mathbf{x}}\tilde{\mathbf{y}}^{\mathrm{T}}\right] \\ &= \begin{bmatrix} \mathrm{E}\left[\tilde{x_1}\tilde{y_1}\right] & \cdots & \mathrm{E}\left[\tilde{x_1}\tilde{y_n}\right] \\ \vdots & \ddots & \vdots \\ \mathrm{E}\left[\tilde{x_m}\tilde{y_1}\right] & \cdots & \mathrm{E}\left[\tilde{x_m}\tilde{y_n}\right] \end{bmatrix}.\end{aligned}$$

The vector variance follows in the usual fashion.

$$\mathrm{Var}\left(\mathbf{x}\right) = \mathrm{Cov}\left(\mathbf{x},\mathbf{x}\right)$$

From this, the variance of a concatenated vector can be written in terms of the concatenation of the covariances of its constituent vectors.

$$\begin{aligned}\mathrm{Var}\left(\begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_n} \end{bmatrix}\right) &= \begin{bmatrix} \mathrm{Cov}\left(\mathbf{x_1},\mathbf{x_1}\right) & \cdots & \mathrm{Cov}\left(\mathbf{x_1},\mathbf{x_n}\right) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left(\mathbf{x_n},\mathbf{x_1}\right) & \cdots & \mathrm{Cov}\left(\mathbf{x_n},\mathbf{x_n}\right) \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{Var}\left(\mathbf{x_1}\right) & \cdots & \mathrm{Cov}\left(\mathbf{x_1},\mathbf{x_n}\right) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left(\mathbf{x_n},\mathbf{x_1}\right) & \cdots & \mathrm{Var}\left(\mathbf{x_n}\right) \end{bmatrix}\end{aligned}$$

II-3 Directed graph representation of s-circuits

When considering an s-circuit, it is helpful to visualize the flow of information among its constituent modules and datapaths. Figure 2 presents the directed graph (digraph) representation of an arbitrary s-circuit:

- *Nodes* represent modules. The labels attached to them name the relational functions that map each module's inputs to its output.

- *Dark arrows* represent datapaths that send information in the direction specified. The labels attached to them name the relational functions that map each datapath's singular input to its output. If a datapath is unlabeled, it is assumed to directly map its input to its output.

- *Light arrows* designate source and output modules. The labels attached to them name the outputs of the s-circuit. Figure 2 depicts an s-circuit with three inputs and two outputs.

- *Labels* name the relational functions that define the elements. It is important to note that the relational functions might be random; a given element might introduce noise into the data it operates upon.

Note that the relational functions may be random. A given module or datapath may introduce additional noise to the data that pass through it; for example, a datapath representing a radio link might introduce white noise into the signal that is sent over it.

The digraph representation is not a complete description of an s-circuit because it ignores the embedding of the circuit within its environment. If the relational functions and input variances can be determined for a specific embedding, then the s-circuit can be analyzed given that embedding. In many types of s-circuit, it may be possible to build a general characterization of the behavior of an s-circuit across a wide range of embeddings.

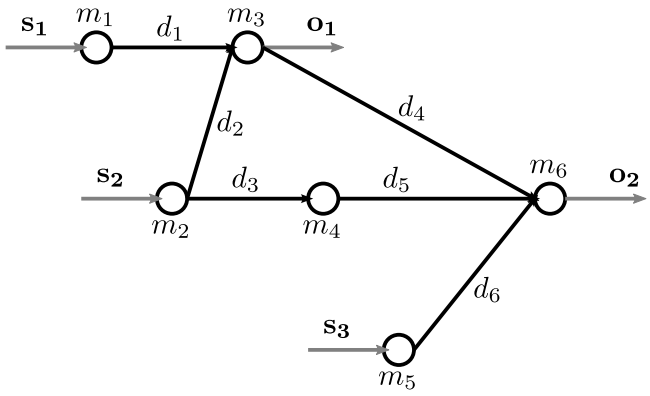


Figure 2: A representational s-circuit in digraph form.

From the digraph representation, each output of the s-circuit can be described using the composition of functions. For the s-circuit of Figure 2, these compositions are:

$$\mathbf{o_1} = m_3\big(d_1 \circ m_1(\mathbf{s_1}), d_2 \circ m_2(\mathbf{s_2})\big)$$
$$\mathbf{o_2} = m_6\Big(d_4 \circ m_3\big(d_1 \circ m_1(\mathbf{s_1}), d_2 \circ m_2(\mathbf{s_2})\big), d_5 \circ m_4 \circ d_3 \circ m_2(\mathbf{s_2}), d_6 \circ m_5(\mathbf{s_3})\Big)$$

By inspection, it is not possible to describe an s-circuit in this way unless its digraph representation is a directed acyclic graph (DAG); for this reason, we will restrict our analysis to s-circuits that can be represented by a DAG. An alternative way to state this restriction is that it must be possible to introduce a topological ordering on the modules of s-circuit's digraph. However stated, this is a powerful restriction on the class of s-circuits that we will consider — it precludes any sort of feedback loop.

From this compositional representation, it should be possible to determine the complete probability density function of the s-circuit's outputs in terms of the inputs and the relational functions. However, this approach would require an unreasonable amount of infor-

mation — the complete probability density functions of the input vectors, as well as full knowledge of each relational function. To step around these requirements, we will develop upon the results of Smith *et al.* [9] to produce an estimate of the output variances of an s-circuit.

II-4 Uncertainty in general linear relationships

Smith *et al.* develop a formula for the variance of a general linear relational function, equation (2) (reproduced here) [9].

$$\mathrm{Var}\left(f(\mathbf{x})\right) = \left[\frac{\partial f}{\partial \mathbf{x}}\right] \mathrm{Var}\left(\mathbf{x}\right) \left[\frac{\partial f}{\partial \mathbf{x}}\right]^{\mathrm{T}}$$

As it is, this result is not general enough to be applied to analysis of s-circuits. It considers only deterministic, univariate (with respect to a single vector) relational functions, while s-circuits may contain random, multivariate relational functions. Sections II-4 and II-4 address these shortcomings.

Validity of linearization

Before proceeding, it will be prudent to provide justification for the widespread use of linear approximations around the mean when estimating variance. To begin with, note that the variance through the linear approximation of a function will rarely be the true variance (this will occur only if the function is itself a linear function). This occurs because the variance is (by definition) a quantity that occurs in the region around the mean, not just at the mean.

Figures 3(a) and 3(b) illustrate the projection of an interval through a function and its linear approximation about the center of the interval. Because the first-order Lagrange Error Bound provides a worst-case bound on the difference between the function and its linear approximation, the maximum error in the width of the projected interval can be expressed as

$$width\ error \leq 2 \cdot Lagrange\ Error\ Bound = \left|\frac{\partial^2 f}{\partial x^2}\right|_{max} |x - c|^2$$

where

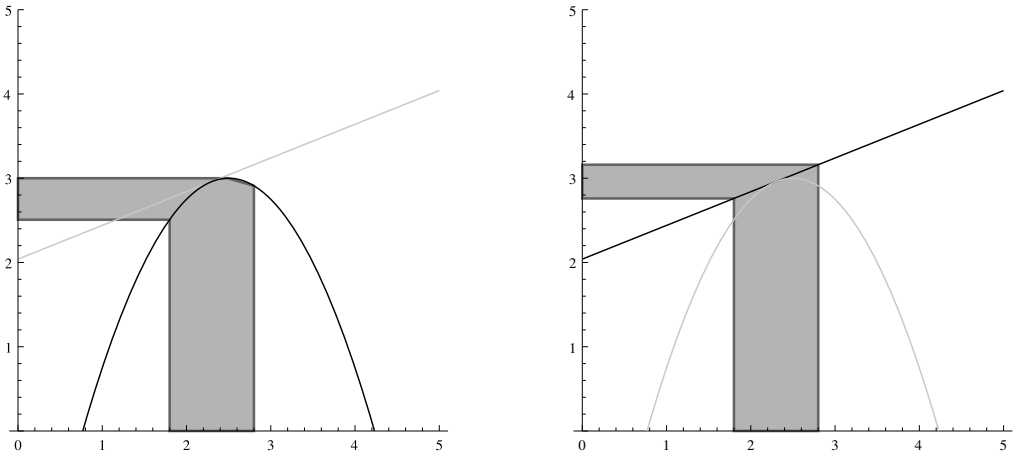
$\left|\frac{\partial f^2}{\partial^2 x}\right|_{max} \equiv$ The maximum absolute value of the second derivative between x and c.

If one considers the interval of 3 to represent a sample distribution consisting of two points, then it is clear that the variance of the linear approximation can be considered to closely follow the true variance in the region where the Lagrange Error Bound is small, or equivalently, when the variance is small relative to the curvature of the true function.

Finally, as Smith *et al.* note [9], equation (2) makes no assumption about the distribution of the random variable $\mathbf{x}$. In many situations, it may be useful to consider normal random variables because their highest moment is variance, but variables of any distribution may be used as input variables without loss of accuracy.

Multivariate relational functions

Multivariate relational function map more than one input vector to their output vector. For example, m_6 in Figure 2 maps the results of d_4, d_5, and d_6 to its output. Multivariate relational functions can be fit into the form required by equation (2) using vector concatenation



(a) Projection through the function. (b) Projection through the linear approximation.

Figure 3: Projection of an interval through a function and its linear approximation about the center of the interval. Note that the projected intervals are not equivalent.

of their arguments.

$$\mathbf{y} = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_n} \end{bmatrix}$$

$$\begin{aligned} \mathrm{Var}\left(f(\mathbf{x_1}, \cdots, \mathbf{x_n})\right) &= \mathrm{Var}\left(f(\mathbf{y})\right) \\ &= \left[\frac{\partial f}{\partial \mathbf{y}}\right] \mathrm{Var}\left(\mathbf{y}\right) \left[\frac{\partial f}{\partial \mathbf{y}}\right]^{\mathrm{T}} \\ &= \left[\frac{\partial f}{\partial \mathbf{y}}\right] \begin{bmatrix} \mathrm{Var}\left(\mathbf{x_1}\right) & \cdots & \mathrm{Cov}\left(\mathbf{x_1}, \mathbf{x_n}\right) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left(\mathbf{x_n}, \mathbf{x_1}\right) & \cdots & \mathrm{Var}\left(\mathbf{x_n}\right) \end{bmatrix} \left[\frac{\partial f}{\partial \mathbf{y}}\right]^{\mathrm{T}} \end{aligned}$$

The variance submatrices in the above matrix are well-defined — their values can be determined from knowledge of the s-circuit's inputs and the function compositions through which they have passed. The covariance submatrices, on the other hand, require a fuller

characterization of the s-circuit than can be expected to available. In general, unless an explicit dependence can be seen in the digraph (see section II-4), it will be assumed that the there is no covariance between an input pair:

$$\mathrm{Var}\left(f(\mathbf{x_1},\cdots,\mathbf{x_n})\right) = \left[\frac{\partial f}{\partial(\mathbf{x_1},\cdots,\mathbf{x_n})}\right] \begin{bmatrix} \mathrm{Var}\left(\mathbf{x_1}\right) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathrm{Var}\left(\mathbf{x_n}\right) \end{bmatrix} \left[\frac{\partial f}{\partial(\mathbf{x_1},\cdots,\mathbf{x_n})}\right]^{\mathrm{T}} \tag{3}$$

Note the omission of the dummy variable $\mathbf{y}$.

If there is reason to suspect that two module input vectors $\mathbf{x}$ and $\mathbf{y}$ are covariant even though no explicit dependence can be seen in the s-circuit's digraph, a standard result known as the Cauchy-Schwarz identity,

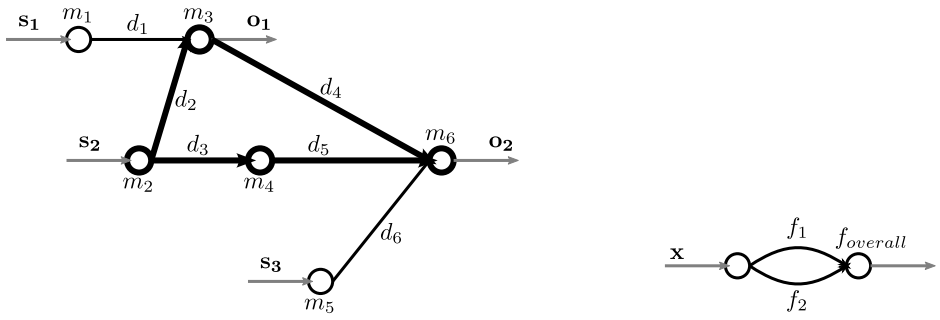
$$\left|\mathrm{Cov}\left(a,b\right)\right| \leq \sqrt{\mathrm{Var}\left(a\right)\mathrm{Var}\left(b\right)}, \tag{4}$$

can be used to provide extremum bounds on the individual scalar elements of the covariance submatrix given by $\mathrm{Cov}\left(\mathbf{x},\mathbf{y}\right)$.

Caveat: explicitly dependent inputs

Explicitly dependent inputs appear in situations such as the one highlighted in Figure 4(a), in which two module inputs are dependent on the same source vector. In this case, it is possible to explicitly compute the value of the corresponding off-diagonal sub-matrix.

To illustrate, consider the degenerate case of explicit dependence, illustrated in Figure 4(b), in which two inputs to a module are simply different functions of the same input. Note that



(a) Representative s-circuit digraph with explicit input dependence marked.

(b) Degenerate case of explicit input dependence.

Figure 4: Examples of explicit input dependence.

the dependence in Figure 4(a) is equivalent to that of 4(b) if we let $f_1 = d_4 \circ m_3 \circ d_2 \circ m_2$ and $f_2 = d_5 \circ m_4 \circ d_3 \circ m_2$, where $\circ$ denotes functional composition. To compute the corresponding off-diagonal sub-matrix, start from the definition of vector covariance:

$$\mathrm{Cov}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right) = \mathrm{E}\left[(f_1(\mathbf{x}) - \mathrm{E}\left[f_1(\mathbf{x})\right])(f_2(\mathbf{x}) - \mathrm{E}\left[f_2(\mathbf{x})\right])^{\mathrm{T}}\right].$$

Replace f_1 and f_2 with their first-order Taylor expansions, noting that $\mathrm{E}\left[f\mathbf{x}\right] = f(\bar{\mathbf{x}})$:

$$\mathrm{Cov}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right) \approx \mathrm{E}\left[\left(\left[\frac{\partial f_1}{\partial \mathbf{x}}\right](\mathbf{x} - \bar{\mathbf{x}})\right)\left(\left[\frac{\partial f_2}{\partial \mathbf{x}}\right](\mathbf{x} - \bar{\mathbf{x}})\right)^{\mathrm{T}}\right].$$

This expression can be manipulated to yield the result

$$\mathrm{Cov}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right) \approx \left[\frac{\partial f_1}{\partial \mathbf{x}}\right] \mathrm{E}\left[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^{\mathrm{T}}\right] \left[\frac{\partial f_2}{\partial \mathbf{x}}\right],$$

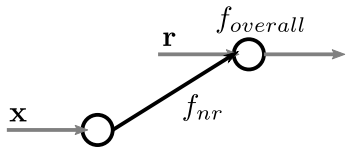


Figure 5: Digraph expansion of stochastic relational function.

which is exactly:

$$\mathrm{Cov}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right) \approx \left[\frac{\partial f_1}{\partial \mathbf{x}}\right] \mathrm{Var}\left(\mathbf{x}\right) \left[\frac{\partial f_2}{\partial \mathbf{x}}\right]. \tag{5}$$

Equation (5) is the linear approximation for the off-diagonal submatrix corresponding to $\mathrm{Cov}\left(f_1(\mathbf{x}), f_2(\mathbf{x})\right)$.

Stochastic relational functions

Because we are concerned only with estimates of variance, it is sufficient to determine the additional variance that a stochastic relational function introduces. Our analysis will be confined to relational functions of the fashion

$$f(\mathbf{x}) \approx f_{nr}(\mathbf{x}) + \mathbf{r} \tag{6}$$

in which the stochastic relational function f can be approximated by a deterministic function f_{nr} plus an additive random vector $\mathbf{r}$. Decompositions of this form are suitable for (simplistic) modeling of many noisy processes, such as additive white noise.

Note that this decomposition describes an s-circuit with digraph given by Figure 5, in

which the multivariate relational function $f_{overall}$ adds together its two inputs. Thus, it is possible to apply the methods just developed for the analysis of multivariate relational functions.

$$\begin{aligned}
\mathbf{y} &= \begin{bmatrix} f_{nr}(\mathbf{x}) \\ \mathbf{r} \end{bmatrix} \\
\mathrm{Var}\left(f_{overall}(\mathbf{y})\right) &= \left[\frac{\partial f_{overall}}{\partial y}\right] \mathrm{Var}\left(\mathbf{y}\right) \left[\frac{\partial f_{overall}}{\partial y}\right]^{\mathrm{T}} \\
&= \left[\frac{\partial f_{overall}}{\partial y}\right] \begin{bmatrix} \mathrm{Var}\left(f_{nr}(\mathbf{x})\right) & \mathrm{Cov}\left(f_{nr}(\mathbf{x}), \mathbf{r}\right) \\ \mathrm{Cov}\left(\mathbf{r}, f_{nr}(\mathbf{x})\right) & \mathrm{Var}\left(\mathbf{r}\right) \end{bmatrix} \left[\frac{\partial f_{overall}}{\partial y}\right]^{\mathrm{T}}
\end{aligned}$$

Note that

$$\left[\frac{\partial f_{overall}}{\partial \mathbf{y}}\right] = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

for any choice of $\mathbf{y}$. The quantities $f_{nr}(\mathbf{x})$ and $\mathbf{r}$ are not covariant, so the covariance submatrices above are $\mathbf{0}$.

$$\begin{aligned}
\mathrm{Var}\left(f_{overall}(\mathbf{y})\right) &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} \mathrm{Var}\left(f_{nr}(\mathbf{x})\right) & \mathbf{0} \\ \mathbf{0} & \mathrm{Var}\left(\mathbf{r}\right) \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \\
&= \mathrm{Var}\left(f_{nr}(\mathbf{x})\right) + \mathrm{Var}\left(\mathbf{r}\right)
\end{aligned}$$

Applying equation (2) again yields

$$\mathrm{Var}\left(f_{overall}(\mathbf{y})\right) = \left[\frac{\partial f_{nr}}{\partial \mathbf{x}}\right] \mathrm{Var}\left(\mathbf{x}\right) \left[\frac{\partial f_{nr}}{\partial \mathbf{x}}\right]^{\mathrm{T}} + \mathrm{Var}\left(\mathbf{r}\right).$$

Thus, for any stochastic relational function f that can be approximated by a deterministic

function f_{nr} with additive noise $\mathbf{r}$,

$$\mathrm{Var}\left(f(\mathbf{x})\right) \approx \left[\frac{\partial f_{nr}}{\partial \mathbf{x}}\right] \mathrm{Var}\left(\mathbf{x}\right) \left[\frac{\partial f_{nr}}{\partial \mathbf{x}}\right]^{\mathrm{T}} + \mathrm{Var}\left(\mathbf{r}\right). \tag{7}$$

II-5 S-Circuit variance estimates

Using the results of section II-4, it is possible to define an algorithm for estimating the output variance of an s-circuit, given:

- That the s-circuit's digraph is a DAG (it contains no cycles)
- That the variance of each input to the s-circuit is known, and
- That the linear approximation of each relational function (including stochastic relational functions, section II-4), is known.

The fact that the s-circuit digraph is a DAG allows us to impose a topological ordering on the elements ensures that each element will be visited before any of its dependents, allowing a single pass through the graph to determine the variance of the s-circuit's outputs.

Algorithm 1 takes as input a DAG G that represents an s-circuit. To simplify the algorithm, this graph is assumed to have a different form than the previous examples of s-circuit digraphs. Both modules and datapaths are represented by nodes of the s-circuit, which have the following fields:

inputs Set of input elements to this element's relational function. Datapaths have only one input.

outputs Set of elements to which the output of this element's relational func-

tion is sent. Datapaths have only one output.

fnr The Jacobian matrix of the deterministic part of this element's relational function, as discussed in section II-4.

fr Variance of the random part of this element's relational function, as discussed in section II-4.

var Variance matrix for this element's output. If this element is a source module for the s-circuit, then this field will be supplied, otherwise it is calculated as the algorithm runs.

depend A map of modules to Jacobian matrices used for detecting and correcting explicit input dependence. Each key is an s-circuit source module on which this element depends, and each value is the Jacobian matrix for the composite linear transform from the key module to this node.

The edges of the expanded digraph G carry no information apart from indicating the dependence of s-circuit elements on other elements. This expansion does not change the DAG property of the original s-circuit graph. Figure 6 shows the expanded form of the representational s-circuit of Figure 2.

The algorithm proceeds by imposing a topological ordering on G, an act that partitions the DAG into a number of equivalence classes. The order in which the nodes of a given equivalence class must be visited is irrelevant, but each must be visited before any nodes of the next equivalence class may be visited.

For each node N, the following steps are taken:

1. Inspect the **var** field of N. If it is already filled out, then N is an input module, and

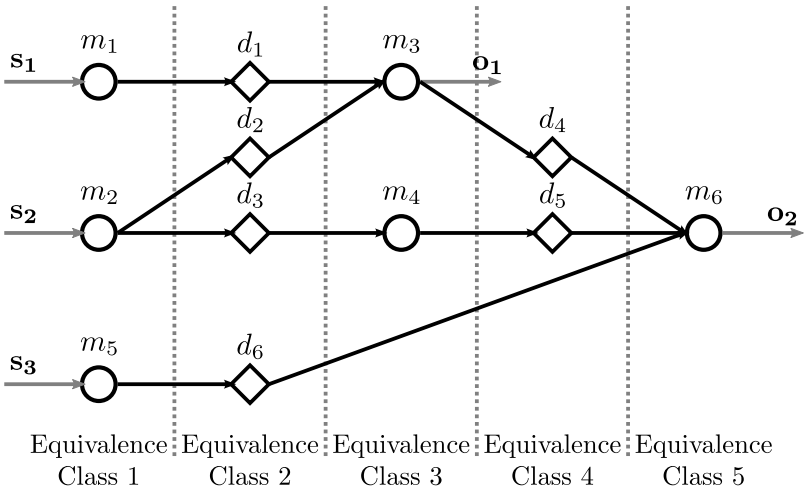


Figure 6: The digraph of Figure 2 after expansion and the imposition of a topological ordering. The equivalence classes of the topological ordering are indicated.

no action should be taken for it.

2. Create an input variance matrix V for N. The diagonal blocks of V are the **var** fields of each input node to N.

3. Inspect each unique pair (P, Q) of inputs to N. If $P.depend$ and $Q.depend$ share a set of keys I, then P and Q are explicitly dependent inputs to N. Correct the submatrix of V that corresponds to (P, Q) by assigning it the value

$$\sum_{J \in I} (P.depend[J])(J.var)(Q.depend[J])^{\mathrm{T}},$$

which is the sum over all shared sources between P and Q of the explicit dependence correction, equation (5). Assign the (Q, P) submatrix the transpose of the (P, Q) submatrix.

4. Calculate the output variance of N according to equation (7) using $N.fnr$, V, and $N.fr$.

5. Update the keys of **depend** by taking the union of all **depend** fields of N's inputs (N depends on every module that its inputs depend on). Update the values of **depend** by left multiplying the relevant sections of $N.fnr$ with the column concatenation of the **fnr** fields of the relevant inputs to N.

Once every node has been visited, the output variance estimates may be read out of the **var** fields of the s-circuit's output modules.

Algorithm 1 has a nontrivial running time for large graphs because of the complicated logic for determining and correcting explicit input dependence. A cursory inspection by counting the looping operations would indicate that it is $\in O(V^3)$, where V is the number of nodes in the graph. However, it is most likely of a lower order, as there are complex interactions between the number of elements traversed at each loop level. However, the main focus of this work is verifying the validity of this approach, not evaluating its efficiency.

II-6 Examples

Beacon-compass reconfiguration

The beacon-compass reconfiguration given by Donald [2] is characterized by the digraph depicted in Figure 7(a), with the relational function

$$\theta_R\left(B_R, H_A\right) = \pi + B_R - H_A.$$

Algorithm 1 An algorithm for determining the output uncertainty of a DAG-represented s-circuit.

Input: s-circuit expanded DAG G

Output: output nodes of G contain variance estimates

1: $O = \mathrm{tsort}(G)$
2: **for all** $N \in O$ **do**
3: **if** $N.var$ not defined **then**
4: matrix $V = \begin{bmatrix} M.inputs[1].var & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & M.inputs[n].var \end{bmatrix}$
5: **for all** combinations $(P, Q) \in N.inputs, P \neq Q$ **do**
6: $I = P.depend \cap J.depend$
7: **if** $I \neq \emptyset$ **then**
8: matrix $S = \mathbf{0}$
9: **for all** $J \in I$ **do**
10: $S = S + P.depend[J] \times J.var \times (Q.depend[J])^{\mathrm{T}}$
11: **end for**
12: $((P, Q)$ submatrix of $V) = S$
13: $((Q, P)$ submatrix of $V) = S^{\mathrm{T}}$
14: **end if**
15: **end for**
16: $N.var = N.fnr \times V \times (N.fnr)^{\mathrm{T}} + N.fr$
17: **for all** $P \in N.inputs$ **do //** Update the keys of **depend**
18: $N.depend = N.depend \cup P.depend$
19: **end for**
20: **for all** $P \in N.depend$ **do //** Update the values of **depend**
21: list $L = \emptyset$
22: **for all** $Q \in N.inputs$ **do**
23: **if** $P \in Q.depend$ **then**
24: $L = L \cup Q$
25: **end if**
26: **end for**
27: matrix $J =$ column concatenation of $(M \in L).depend[P]$
28: matrix $K =$ row concatenation of L-respective submatrices of $N.fnr$
29: $N.depend[P] = K \times J$
30: **end for**
31: **end if**
32: **end for**

(a) Beacon-Compass reconfiguration. (b) Beacon-Only reconfiguration.

Figure 7: Digraph representations of Donald's beacon-compass reconfiguration and beacon-only reconfiguration.

This is a straightforward digraph with no explicit dependence, so equation (3) may be directly applied:

$$\begin{aligned}
\mathrm{Var}\left(\theta_R\right) &= \left[\frac{\partial \theta_R}{\partial (B_R, H_A)}\right] \mathrm{Var}\left(\begin{bmatrix} B_R \\ H_A \end{bmatrix}\right) \left[\frac{\partial \theta_R}{\partial (B_R, H_A)}\right]^{\mathrm{T}} \\
&= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \mathrm{Var}\left(B_R\right) & 0 \\ 0 & \mathrm{Var}\left(H_A\right) \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\
&= \mathrm{Var}\left(B_R\right) + \mathrm{Var}\left(H_A\right) \qquad (8)
\end{aligned}$$

In other words, we expect the variance of the relative bearing to the goal to be the sum of the variance of the compass and the variance of the beacon.

Beacon-only reconfiguration

The beacon-only reconfiguration given by Donald [2] is characterized by the digraph depicted in Figure 7(b). Using function composition, the relative bearing to the goal can be

written as

$$\theta_R = f_{comm}\left(B_R\right).$$

The function f_{comm} is assumed to be a stochastic function that can be decomposed into a deterministic function f_{nr} and an additive noise term r. For specificity, an arbitrary choice of deterministic function

$$f_{nr}(B_R) = aB_R$$

is made. Applying the formula for the variance of a stochastic function, equation (7), yields

$$\begin{aligned}\operatorname{Var}\left(\theta_R\right) &= \left[\frac{\partial f_{nr}}{\partial B_R}\right] \operatorname{Var}\left(B_R\right) \left[\frac{\partial f_{nr}}{\partial B_R}\right]^{\mathrm{T}} + \operatorname{Var}\left(r\right) \\ &= a^2 \operatorname{Var}\left(B_R\right) + \operatorname{Var}\left(r\right) \end{aligned} \tag{9}$$

In other words, we expect the variance of the relative bearing to the goal to be the sum of the scaled variance of the beacon and the variance of the stochastic element of the communications channel.

CHAPTER III

EMPIRICAL CONFIRMATION PROCEDURE

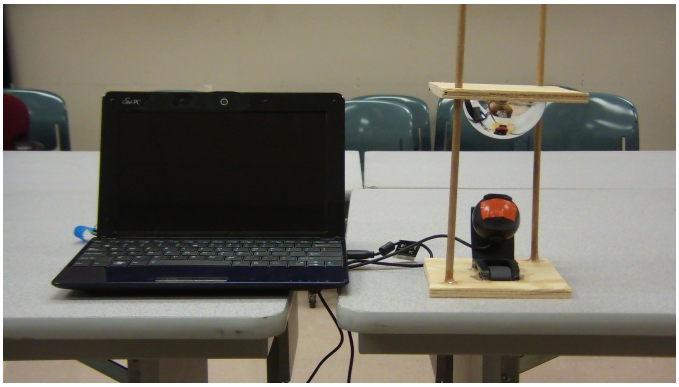
The variance estimation procedure constructed in section II-5 was tested on two of the minimal task sensor reconfigurations due to Donald [2] — the beacon-compass reconfiguration and the beacon-only reconfiguration. The implementations of these s-circuits are discussed in section III-1, and the statistical test used is discussed in section III-2.

III-1 Implementation

Beacon

The beacon consists of two separate pieces of hardware, the emitter and the receiver. The emitter, shown in Figure 8(a), has a directional, rotating red light, made from a red party light with a paper hood to restrict the the output to a narrow beam. Affixed to the top of the red light's housing is a bank of high-intensity blue LEDs, controlled by a magnetic reed switch that is triggered when the red beam passes over it. The receiver, shown in Figure 8(b), consists of a Logitech C120 USB camera pointed upwards at a silvered ornament, to give a wide-angle polar view of its surroundings. The camera is controlled by software running on an Ubuntu GNU/Linux Asus Eee PC 1005HA. Because the camera is operating in a darkened environment, it was necessary to disable auto-exposure using the `guvcview` software package to keep the camera from slowing its framerate in order to gather more light per frame. Additionally, the camera's focus was manually adjusted to be very short, causing the beacon to appear as bright blobs in the camera feed, rather than sharply defined points of light.

(a) Emitter module.



(b) Receiver module.

Figure 8: The hardware implementations of the beacon emitter and receiver.

Red and blue lights were chosen over the green and white lights of Donald because the camera reports its image in YUV colorspace (luminance, red chrominance, and blue chrominance). Taking advantage of this hardware separation greatly reduces the software processing that must be performed on the camera feed. Once a frame has been loaded and unpacked, the average red and blue content are computed. If the red content or blue content goes above a threshold value, then the respective light is considered to be visible, or in a peak. The time between blue and red peaks is measured, and then used to compute the bearing from the beacon to the robot B_R.

The beacon does use a continually-updated estimate for the average period of the rotating light. This results in a fixed start-up cost as the estimate of B_R converges to its true value.

Compass

The compass, shown in Figure 9, is a Devantech CMPS03 magnetic compass module, which provides both I2C and analog readout formats. A Devantech USB-I2C bridge is used to interface the compass with software running on an Ubuntu GNU/Linux Asus Eee PC 1005HA. No software processing or filtering is performed on the raw compass read-

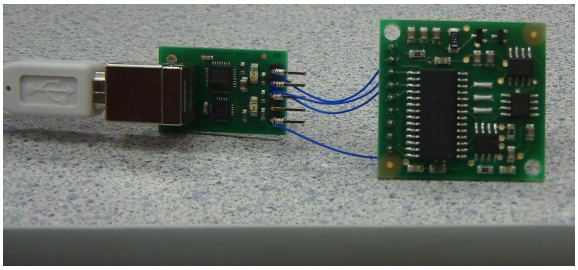


Figure 9: The hardware implementation of the compass.

ings. The overall experimental setup is shown in Figure 10.

Beacon-compass reconfiguration

The beacon-compass reconfiguration was set up with the compass attached to the beacon receiver module, and the beacon emitter's local North aligned with magnetic North. The compass/receiver and emitter were placed at separate ends of a 1.8 meter table, under the configurations listed in Table 1. The lights were turned out to minimize the appearance of red and blue in the environment, and the beacon and capture software was run for 10000 frames, which corresponds to approximately 5 minutes. Despite the disabling of auto-exposure, the framerate still exhibited some variability, so measuring capture length by time is not exact. Every time that a red peak was detected after a blue peak (and a measurement of B_R became available) various measurements, including the current time, beacon reading, compass reading, and relative heading to goal were recorded.

Beacon-only reconfiguration

The beacon-only reconfiguration was tested as an off-line simulation on top of the beacon readings B_R from the beacon-compass reconfiguration, simulating equation (9) with parameters $a = 2$ and $\mathrm{Var}\,(r) = 10$. Wolfram Mathematica 8 was used to apply the

Table 1: Embedding parameter values for the beacon-compass reconfiguration.

Trial	B_R (°)	H_A (°)
1	60	0
2	60	90
3	60	180
4	60	270
5	240	0
6	240	90
7	240	180
8	240	270

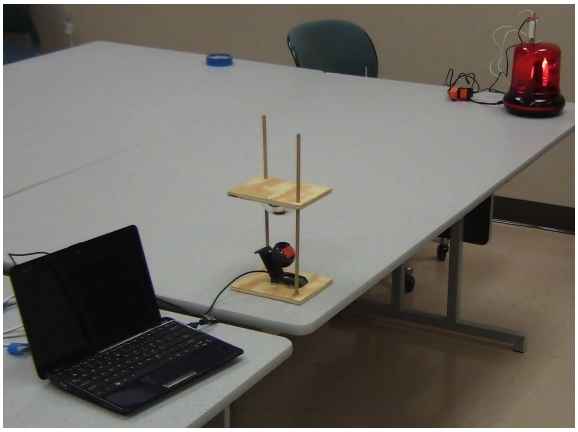


Figure 10: Experimental setup for the beacon-compass reconfiguration.

scale factor to B_R and add a vector of normally-distributed pseudorandom numbers with parameters $\mu = 0$ and $\sigma^2 = 10$.

The choice of an off-line simulation was made due to the difficult set-up and low-noise characteristics of most available communications channels. It would have been possible to introduce noise artificially in the software transmitting the data, but this situation would be no different from adding the noise after the fact.

III-2 Statistical test

The purpose of these experiments is to make inferences about the variance of a population from a set of sample variances. Most common statistical tests are focused on inferences about means. Fortunately, sets of m sample variances of a population are themselves (nearly) normally distributed for large sample sizes ($n \geq 100$) [7]:

$$SampleVariance \sim \mathrm{N}\left(\mu_{var} = \frac{(n-1)\sigma^2}{n}, \sigma^2_{var} = \frac{2\sigma^4}{n}\right).$$

Note that the mean goes to σ^2 in the limit as n goes to infinity. With this knowledge, it is possible to recast this problem as a T-test with $m - 1$ degrees of freedom, with the test statistic

$$t = \frac{\mathrm{mean}(s^2) - \sigma^2}{\sqrt{\frac{\mathrm{Var}(s^2)}{n}}}.$$

The usual test can then be performed, with null hypothesis $H_= : \sigma^2 = predicted\ value$ and $H_{\neq} : \sigma^2 \neq predicted\ value$. Large p-values will provide evidence for not rejecting the null hypothesis in favor of the alternative hypothesis. All data analysis is performed using

Wolfram Mathematica 8.

CHAPTER IV

RESULTS

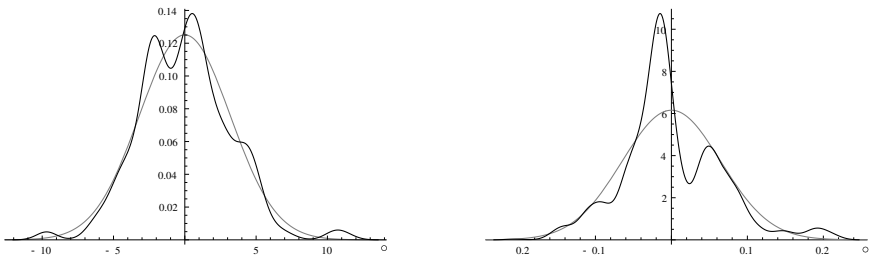
IV-1 Beacon-compass reconfiguration

Each of the $m = 8$ test configurations produced a data vector. To account for the fixed start-up cost of the beacon, the first 100 data points of each set were truncated, and then each data set was truncated to the length of the shortest set. This resulted in the value $n = 130$ for the size of each data sets.

To obtain the variance estimates of B_R required by equation (8), the residuals of B_R across all data sets were grouped together and their sample variance $\mathrm{Var}\,(B_R) = 10.172^{\circ^2}$ was measured. The same procedure applied to H_A yielded $\mathrm{Var}\,(H_A) = 0.004^{\circ^2}$. Because these estimates were taken from many data points spread across many permutations of the s-circuit, they will be treated as given quantities. Figure 11 compares the smooth histograms of B_R and H_A to normal distributions with the same variances. The variance prediction of equation (8) is thus

$$\mathrm{Var}\,(B_R) + \mathrm{Var}\,(H_A) =$$
$$10.172^{\circ^2} + 0.004^{\circ^2} = 10.177^{\circ^2}.$$

The sample variances of the relative bearing to goal θ_R in each data set are recorded in Table 2, and their sampling distribution is shown (with the variance predicted by equation (8) superimposed) in Figure 12(a). Performing the test outlined in section III-2 yielded the test statistic $t = 0.017$ corresponding to the p-value $p = 0.987$.

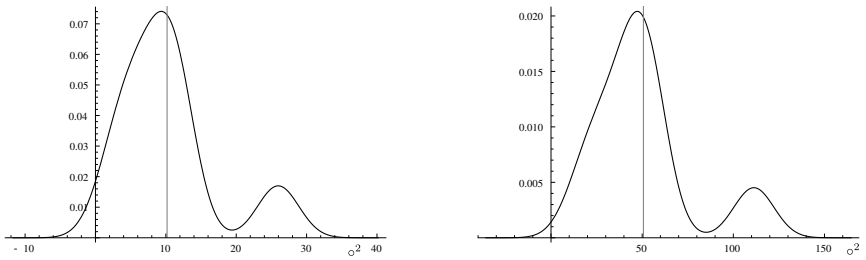


(a) Beacon. (b) Compass.

Figure 11: Smooth histograms comparing the residuals of the beacon (B_R) and the compass (H_A) to normal distributions with the same variance. The near-normality of this data is not a requirement, and is noted as an aside. Note the large difference in the width of the histograms.

Table 2: Sample variances of θ_R across 8 trials for the beacon-compass reconfiguration.

Trial	Sample Variance ($^{\circ 2}$)
1	10.922
2	4.557
3	6.833
4	11.178
5	11.676
6	25.971
7	8.133
8	2.491



(a) Beacon-Compass reconfiguration. The predicted variance is due to equation (8).

(b) Beacon-Only reconfiguration. The predicted variance is due to equation (9).

Figure 12: Smooth histograms of $\mathrm{Var}\,(\theta_R)$ across all 8 data sets for both reconfigurations. The gray vertical lines mark the variance predicted by the relevant predictor equation.

Table 3: Sample variances of θ_R across 8 trials for the beacon-only reconfiguration.

Trial	Sample Variance ($^{\circ 2}$)
1	54.798
2	31.295
3	39.852
4	49.375
5	58.556
6	119.974
7	39.329
8	18.400

IV-2 Beacon-only reconfiguration

The values of B_R measured in the previous section were reused to perform the simulation of the beacon-only reconfiguration discussed in III-1. For this experiment, the variance prediction of equation (9) is thus

$$a^2 \operatorname{Var}(B_R) + \operatorname{Var}(r) =$$
$$4 \cdot 10.172^{\circ 2} + 10^{\circ 2} = 50.690^{\circ 2}$$

The sample variances of the relative bearing to the goal θ_R are recorded in Table 3. The sampling distribution of θ_R is shown (with the variance predicted by equation (9) superimposed) in Figure 12(b). Performing the test outlined in section III-2 yielded the test statistic $t = 0.070$ corresponding to the p-value $p = 0.946$.

CHAPTER V

DISCUSSION AND SUMMARY

V-1 Experimental results

The histograms of Figure 11 illustrate the general output distribution of the beacon (B_R) and the compass (H_A). It is interesting to note that these distributions are close to normal distributions, though this is by no means a requirement for the variance estimation procedure. The finite framerate of the beacon receiver's camera is reflected in Figure 11(a) by the irregularities around $\pm 6^\circ$ and $\pm 12^\circ$. These irregularities represent a form of quantization error — the camera's framerate (approximately 30 frames per second) and the beacon's period (approximately 1.785 seconds per rotation) mean that the beacon has an angular granularity of approximately 6°. The readings are not sharply divided between these bins because the beacon uses a continually-updated estimate of the beacon's period. The fact that this estimate changes from reading to reading smears the data between the expected bins, which was not an anticipated effect. The compass exhibits a similar quantization effect that produces artifacts at 0.1° intervals in its histogram, Figure 11(a).

For the beacon-compass reconfiguration, the statistical test provides strong evidence for *not* rejecting $H_=$ ($\mathrm{Var}\,(\theta_R)$ is described by equation (8)) in favor of $H_\neq$ ($\mathrm{Var}\,(\theta_R)$ is not described by equation (8)). In addition, Figure 12(a) shows that the predicted variance is close to the center of the sampling distribution of $\mathrm{Var}\,(\theta_R)$ in this configuration. However, it is possible that the positive results are due to the four order of magnitude difference between $\mathrm{Var}\,(B_R)$ and $\mathrm{Var}\,(H_A)$, since the contribution of $\mathrm{Var}\,(H_A)$ is minimal at most. This possibility casts doubt on the validity of the test. Future work should test a similar s-circuit with more comparable input variances.

The simulation parameters for the beacon-only reconfiguration were chosen to avoid the situation described above. Indeed, the input variances were nearly equivalent. The statistical test performed provides strong evidence for *not* rejecting $H_=$ ($\mathrm{Var}\,(\theta_R)$ is described by equation (9)) in favor of $H_{\neq}$ ($\mathrm{Var}\,(\theta_R)$ is not described by equation (9)). In addition, Figure 12(b) shows that the predicted variance is close to the center of the sampling distribution of $\mathrm{Var}\,(\theta_R)$ in this configuration.

These tests provide verification for the core aspects of the variance estimation procedure given in section II-5, but not some of the more interesting theoretical results. For example, neither the beacon-compass reconfiguration nor the beacon-only reconfiguration exhibit explicit input dependence. In addition, the s-circuits tested were very simple — they contained few nodes, exhibited little chaining or parallelism, and produced only scalar quantities. The s-circuits Donald gives in his work [2] are thus not well-suited to robust experimental verification of the s-circuit variance estimation procedure. Future experimental verification might focus on sensors with a large amount of noise that are easy to combine into a complex systems — for example, ultrasonic rangefinders for position estimation. In addition, sensors producing vector quantities, such as laser rangefinders, might be considered.

V-2 Theoretical results

The variance estimation procedure is sufficient to analyze many types of s-circuits, but suffers from several shortcomings. As previously mentioned, it cannot analyze s-circuits containing feedback loops. Additionally, all analysis of an s-circuit is performed under one embedding of the s-circuit within its environment. Changing the embedding of an s-circuit has the potential to change the means and variances of its inputs as well as the

relational functions of its datapaths. Because the input means determine the Jacobians used to represent the linearizations of relational functions, changing the embedding of an s-circuit can completely invalidate a previous output variance estimate. This problem can be stepped around for the many s-circuits that are reasonably independent of embedding, but a greater accounting for the variability of an s-circuit's parameters is needed.

However, it is important to remember that the advantages of this procedure are its simplicity and generality. If the procedure is made too complex and time-intensive, it will be more efficient to perform explicit simulations of each s-circuit under consideration. In any case, all currently unverified aspects of this procedure should be empirically investigated before considering ways that it might be extended. Particularly, the procedure currently includes a theoretically unfounded assumption at lines 9 and 10. When developing the procedure, the possibly of two or more explicit input dependencies occurring in parallel was not considered. Thus, the algorithm assumes that if this situation occurs, the variances due to each dependency are additive, which has not been theoretically investigated.

V-3 Concluding remarks and future directions

We have extended the theory of s-circuits as presented by Donald [2] to provide a basic estimation of the uncertainty in the output of an s-circuit. Special attention was paid to keeping the information requirements low — no between-component measurements are required, only within-component measurements. Our extension was experimentally verified on the s-circuits given by Donald, with results that tend to confirm the validity of our approach. However, much work still remains, both in experimental verification of the current extension and further theoretical improvements. In particular:

- The current procedure needs to be verified for edge-case s-circuits — s-circuits with

wide, deep, and dense digraphs.

- The current procedure must be verified on s-circuits with pathological features such as significantly nonlinear relational functions and explicit input dependence.

- The current procedure must be extended to cover the case where the parameters of a stochastic function's additive noise are not constant across all embeddings of an s-circuit.

- The current procedure must be developed to properly handle the case where multiple explicit input dependencies on more than one input occur at the same node.

We hope that our future work will allow us to resolve these issues and develop the theory of s-circuits as an invaluable tool in the analysis and optimization of robotic systems.

REFERENCES

[1] J. J. Cox, J. Ingemar, Modeling a dynamic environment using a Bayesian multiple hypothesis approach, Artificial Intelligence 66(2) (1994) 311–344.

[2] B. Donald, On information invariants in robotics, Artificial Intelligence 72(1-2) (1995) 217 – 304.

[3] B. Donald, J. Jennings, Sensor interpretation and task-directed planning using perceptual equivalence classes, in: Proceedings of the IEEE International Conference on Robotics and Automation, volume 1 (1991) pp. 190 –197.

[4] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research 11(3) (1999) 391–427.

[5] R. Negenborn, Robot localization and kalman filters: On finding your position in a noisy world, Master's thesis, Utrecht University (2003).

[6] J. M. O'Kane, S. M. LaValle, Comparing the power of robots, The International Journal of Robotics Research 27(1) (2008) 5–23.

[7] J. J. Schiller, R. A. Srinivasan, M. R. Spiegel, Schaum's Outline of Probability and Statistics, pp. 159–160, McGraw-Hill, $3rd$ edition (2008) .

[8] R. C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, The international journal of Robotics Research 5(4) (1986) 56.

[9] R. C. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics 1 (1990) 167–193.

[10] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press (2005).

CONTACT INFORMATION

Name:	Taahir David Ahmed
Professional Address:	c/o Dr. Dylan Shell Department of Computer Science and Engineering Texas A&M University TAMU 3112 College Station, TX 77843-3112
Email Address:	t551@tamu.edu
Education:	*In Progress:* B.S. Comp. Eng., B.S Physics Texas A&M University, May 2013 Honors Undergraduate Research Fellow